DECLARATION OF INVENTORS UNDER 37 C.F.R. §1.131(a)	Docket No. L0T9-2000-0036

Applicant:	Brian Pulito, Mark Johnson, Brian Cline, Mark S. Kressin, Andrew Lochbaum and Jeff Durham
Serial No:	09/695,193
Filed:	October 24, 2000
For:	METHOD AND APPARATUS FOR PROVIDING FULL DUPLEX AND MULTIPOINT IP AUDIO STREAMING
Examiner:	Oanh L. Duong
Art Unit:	2155

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

The undersigned hereby declares and states:

1. I am a named inventor in the above-identified United States patent application.

2. I have been employed by International Business Machines Corporation, hereafter referred to as IBM, the assignee of record of the above-identified U.S. Patent Application, from 7/1/1999 until the present in the position of Software Engineer

3. Prior to June 28, 2000, the priority date of US Patent 6,683,858, Chu et al., hereafter referred to as Chu, I conceived of at least part of the subject matter disclosed in the above-identified patent application.

4. The inventive concepts were at least partially memorialized in IBM Invention Disclosure LO8-99-0014, entitled "2-Way Distributed Audio Mixing of Multiway Calls," a copy having the date(s) obliterated being attached hereto as Exhibit A and incorporated herein by reference. The contents of Exhibit A establish the fundamental concepts of the invention as recited in each of independent claims 14 and 20. Specifically, Sections 1-4 of the invention disclosure disclose the fundamental concepts described in claim 14, lines 3, *et seq* and claim 20, lines 8, *et seq*.

5. The invention disclosure form was forwarded to IBM Patent counsel, who, in turn, forwarded the invention disclosure to the attorney of record in the above identified patent application, with an email message requesting a patentability search, as also illustrated in Exhibit A.

6. Thereafter, refinement of the inventive concepts continued along with collaboration with the other named inventors in the above identified patent application and a more robust and detailed document produced entitled "MMCU/MMP Design Specification For Watson" describing the details of the inventive concepts, a copy having the date(s) obliterated is attached hereto as Exhibit B and incorporated herein by reference. Specifically, sections 1, 3.1, 3.2 and 4.1 through 4.3 disclose in greater detail support for limitations in the all of claims 14-19 and 20-26.

7. In cooperating with the attorney of record during the preparation of the above-identified patent application, this a copy of Exhibit B was subsequently forwarded to the attorney of record by the electronic-mail communication, a copy having the date(s) obliterated is attached hereto as Exhibit C and incorporated herein by reference.

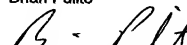
8. Additional miscellaneous electronic mail correspondence between myself and the attorney of record continued during preparation and review of the final patent application draft, a exemplary copy having the date(s) obliterated is attached hereto as Exhibit D and incorporated herein by reference

9. On October 24, 2000, the above-identified patent application which discloses the inventive concept was filed with USPTO and assigned Serial No. 09/695,193.

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of

Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

First Inventor Name: Brian Pulito

Inventor's Signature:  Date: 10/14/2006

Residence: 125 S. Ashland Avenue, Lexington, KY 40502
Citizenship: U.S.A.
Post Office Address: 125 S. Ashland Avenue, Lexington, KY 40502

Second Inventor Name: Mark Johnson

Inventor's Signature: _____ Date: _____

Citizenship: U.S.A.
Residence Address: 763 Dorgene Lane, Cincinnati, OH 45244
Post Office Address: 763 Dorgene Lane, Cincinnati, OH 45244

Third Inventor Name: Brian Cline

Inventor's Signature: _____ Date: _____

Citizenship: U.S.A.
Residence Address: 2078 Woodsedge Court
Post Office Address: Hebron, KY 41048

Inventor's Signature: _____ Date:_____

Citizenship: U.S.A.
Residence Address: 7346 Banbridge Court, West Chester, OH 45069
Post Office Address: 7346 Banbridge Court, West Chester, OH 45069

Fifth Inventor Name: Mark Kressin

Inventor's Signature: _____ Date:_____

Citizenship: U.S.A.
Residence Address: 201 Corinthian, Lakeway, TX 78734
Post Office Address: 201 Corinthian, Lakeway, TX 78734

Sixth Inventor Name: Andrew Lochbaum

Inventor's Signature: _____ Date:_____

Citizenship: U.S.A.
Residence Address: 9816 Lonsdale Drive, Austin, TX 78729
Post Office Address: 9816 Lonsdale Drive, Austin, TX 78729

EXHIBIT
bjobse
A

From: <Steve_Keohane@lotus.com>
To: BK.BK_PO(bjobse)
Date: [redacted] 5:16PM
Subject: LO8-99-0014 - 2-Way Distributed Audio Mixing of Multiway Calls

Hi Bruce,

859

Per our conversation, here is the first of the invention disclosures that need to be searched for patentability. Brian Pulito (~~800~~.425.3564) can answer any questions on this.

bpulito@databeam.com

859-269-7376 H
859-356-7376 alluba
859-425-3528 Fax

I will be sending you another email with eight more inventions, not in this great of detail. On those, Brian will have to fill in the details (or direct you to someone who can).

Liz, this is fyi.

Regards, Steve.

----- Forwarded by Steve Keohane/CAM/Lotus on [redacted] 05:17 PM -----

Invention Disclosure Form

Status

Draft
Lotus Strictly Private
Prepared for Lotus Attorney

General Information

Disclosure No.: LO8-99-0014 (This number will be assigned by Legal)

Title of the Invention: 2-Way Distributed Audio Mixing of Multiway Calls

Cline, Brian

Johnson, Mark

Kressin, Mark

Lochbaum, Andrew

Pulito, Brian

Product/Critical Date Information

1. To which product is this invention related? Sametime
   a. which release? 2.0

2. What is the planned/actual date that this product is to be shipped?

[redacted]

3. What is the planned/actual date that this product is to be announced?

[redacted]

4. Has this product been shipped for beta test (or pre-release)? |---------|
   | ( ) Yes |
(*) No

   a. If so, how many betas were shipped?

5. Has this invention been disclosed outside of Lotus? |---------|
   | (*) Yes |
( ) No

   a. to whom? Marcel Graf/IBM

   b. when? [redacted]

   c. was a CDA in place? |---------|
   | ( ) Yes |
   | ( ) No |
(*) N/A

6. Has a written description of this invention been published outside of Lotus? |---------|
   | ( ) Yes |
(*) No

   a. If yes, when?

   b. where?

Invention Development Information

1. When was the invention first conceived? [redacted]

2. When was the invention first reduced to practice? [redacted]

3. Who other than Lotus are likely to use this invention? Microsoft, Lucent, Nortel, Cisco

4. How likely are others to use this invention now or in the future?

Highly likely

5. How would infringement be discovered? Listening to a multiway IP Audio Call while oberving CPU load on the MCU

Invention Information

1. Description of the problem solved by this invention:

This invention is used to simulate full-duplex audio when more than 2 nodes are participating in a single IP audio call. The invention accomplishes this difficult task in a very efficient manner:

It keeps the CPU load on the MCU at a minimum, which increases the number of simultaneous calls that can be handled by the MCU.

It also keeps network traffic to a minimum.

2. Description of the solution to the problem (summary):

This problem is solved by distributing mixing of the multi-way audio off of the MCU out to the endpoints participating in the call. The Switching MCU's only responsibility is to forward the two most active audio streams to each endpoint participating in the call. This allows every endpoint participating in the call to hear at most two simultaneous speakers. Typically, anymore than two active speakers in a call is illegible anyway so this becomes an acceptable limitation.

3. Description of Advantages over prior art:

This approach has a number of advantages over prior art:

Advantage over Push Button IP Phones - Many IP Audio phones on the market today (Yahoo!, Excite, Centra, etc.) require the participating endpoints to push a button similar to a CB radio before speaking. This informs the MCU to switch to the active stream. This type of technology is a step backward from todays PSTN conference bridges. Obviously, no

push button is required for our invention.

Advantage over Single Stream Switching MCUs - MCUs of this type (such as DataBeam's neT.120 MCU) switch only on a single stream which at its best provides a half duplex multi-way audio call. Clipping of audio when the stream switches from one endpoint to another makes it difficult to provide a natural multiway audio call. Our invention provides for a much more natural way to communicate

Advantage over MCUs that perform Audio Mixing - Centralized mixing MCUs provide very good audio quality at the cost of dedicated hardware. To provide this type of a solution each active stream must be decoded (decompressed), mixed and the mixed stream is then encoded and sent to all endpoints. This requires a major amount of dedicated processing power for every active call which can become cost prohibitive and does not scale well. Because our invention distributes the mixing to the endoints, the MCU does not have to do any transcoding or mixing at the server which allows it to perform all of its responsibilities in software with no dedicated hardware. Our solution also scales reasonably well.

Advantage over PSTN Audio Bridges - This really boils down to advantages of IP telephony over standard POTs audio. No long-distance fees, integration with video and data conferencing etc.

4. Description of the Invention (include diagrams as required):

This invention solves the Multi-way IP Audio problem by opening 2 outgoing audio streams from the MCU to every node participating in a call. Each endpoint is responsible for mixing the two incoming audio streams received from the MCU. Each endpoint sources a single audio stream to the MCU. For this solution to work, the endpoints must support silence suppression (the act of not sending any media to the MCU when silence is detected at the

endpoint). When the MCU detects an active audio stream it "switches" the active stream to all endpoints participating in the call. If a second stream becomes active it is also switched to all the endpoints in the call. The MCU locks on the two active streams until silence is detected (media on one of the streams stops being received by the MCU) and a third stream becomes active. Using this method, at most 2 active streams will be heard in the call. Typically, this is enough for one participant in a call to interrupt a second speaking participant in a natural way without clipping the audio and with some assurance that the speaker will hear the interruption even while talking.

5. Description of any documentation that is relevant to this invention (design documents, etc.):

At this point in time there is no documentation describing this invention other than this Invention Disclosure Form.

Miscellaneous Information

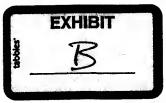
Please provide any other information that you feel is relevant:

Is this invention disclosure complete?

|---------|
| (*) Yes |
( ) No

CC: BK.gwia("Elizabeth_Hart@lotus.com","Brian_Pulito@l...

Lotus Development



MMCU/MMP Design Specification for Watson

Jeff Durham; V1.0.0 ([redacted])

Lotus Development

MMCU/MMP Design Specification for Watson

V1.0.0 ([redacted])

Document Revision History

Date	Version	Author	Revisions / Comments
[redacted]	1.0.0	Jeff Durham	Initial Draft

MMCU/MMP Design Specification for Watson

V1.0.0 ([redacted])

Table of Contents

1 Introduction 1

2 Terminology 2

3 MMCU 5

3.1 Overview 5

3.2 Client Connections 6

3.2.1 SIP 6

3.2.2 H.323 7

3.3 MMP Events 7

4 MMP 7

4.1 Overview 7

4.2 Sequence Numbers 8

4.3 Timestamps 8

4.4 Full Duplex Mode 9

4.5 Two-Way Mixing and Push-To-Talk Mode 9

4.6 Multiple Audio Inbound Streams 9

5 H.323 Gatekeeper Support 10

6 Bandwidth Management 10

6.1 G.723 Audio Codec Bandwidth Usage 10

6.2 G.711 Audio Codec Bandwidth Usage 12

6.3 H.263 Video Codec Bandwidth Usage 12

7 Invited Server 13

8 A/V Tuning Wizard 13

1 Introduction

The Multimedia Control Unit (MMCU) and the Multimedia Processor (MMP) provide multipoint audio and video services for Watson. These server-based components allow users via the Watson Meeting Room Client or an H.323 endpoint to participant in meetings that contain audio and video.

The MMCU/MMP uses a switching technique to provide a multipoint audio and video experience. The MMCU/MMP monitors inbound audio packets from all clients. Once audio packets are received, the MMCU/MMP will lock onto that client and broadcast these packets to other clients in the meeting. After the client sourcing the packets has gone "quiet" for a certain period of time, the MMCU/MMP will then scan for other active audio clients. The MMCU/MMP will lock onto at most two audio input streams. Clients capable of receiving more than one audio stream will receive both of these streams. Those clients are expected to mix the streams before playback to the user. Clients unable to receive both streams such as Microsoft NetMeeting will only receive one of the two streams.

The receipt of audio packets determines audio activity. For this to work properly, silence detection has to be enabled at the client. If a client continued streaming audio packets that only contained silence, the MMCU/MMP would be unable to switch to other clients because it is unaware that the packets only contained silence. In order for the MMCU/MMP to detect silence in the received audio packets, it would have to decode and analyze these packets. This would significantly increase CPU utilization thereby affecting scalability. Overall latency would increase because the audio packet would be delayed before sending it out to the receiving clients.

> The G.723 audio codec has the capability to send a special packet type that indicates silence. These audio packets are not interpreted by the MMCU/MMP as audio activity. They are simply discarded. In other words, even though audio packets are being received from a client, the MMCU/MMP does not switch to this client because it understands that these packets only contain silence.

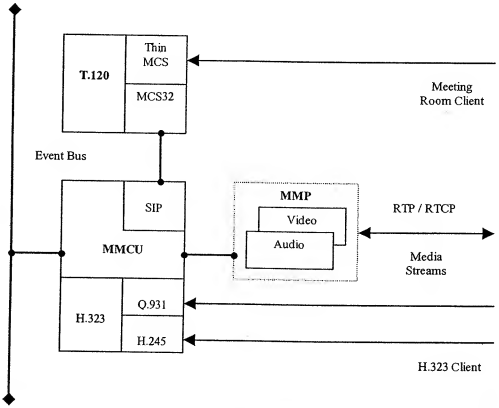
For video, the MMCU/MMP has the video follow the audio. Once the MMCU/MMP has locked onto a client source audio, if that client is also sourcing video, the MMCU/MMP will broadcast that video stream to all other clients. If the active audio client is not capable of sourcing video or has its video stream paused, the receiving clients will be notified that the source video stream is not available. The intent behind this is to allow the receiving client to display information to the user indicating that a video source is unavailable for this active presenter. In order to prevent video from switching too quickly between active presenters, video will lock onto a client for a minimum amount of time.

> As an example, let's say that the minimum video switch time is four seconds. If the active presenter talks for two seconds, then another presenter becomes active for one second, and then the original presenter becomes active again, the video would remain on the original presenter. The second presenter would never have been seen due to the minimum video switch time.

H.323 is an umbrella standard that embodies many other standards for the purpose of audio and video conferencing. These standards provide for such capabilities as call setup, capabilities negotiation, call routing, bandwidth management, and media stream

management and content. The Watson server supports the H.323 standard to the extent that H.323 endpoints and PSTN users via an H.323 gateway can participant in audio and video meetings with other Watson users.

The Watson Meeting Room Client is a "thin client" that utilizes SIP rather than H.323. The Watson server is responsible for providing interoperability between the Meeting Room Client and H.323 endpoints. However, both SIP and H.323 endpoints utilize the same media stream protocol and content. They differ in the way they handle the connection to the server and the setup of the call. H.323 is a large and complex protocol. SIP was chosen for its small size, simplicity to implement, and the speed at which a call can be setup.



2 Terminology

Bandwidth. Bandwidth refers to the amount of data that can flow through your network connection. For instance, if you have a 28.8K modem, then 28.8 kilobits of data is the maximum amount of data that can flow through the connection per second. A single ISDN channel can support a throughput of 64 kilobits of data per second. A typical LAN connection is rated at ten megabits per second although 100 megabits per second is becoming very common. Because of collisions, a 10 Mbps LAN connection can peak at two megabits per second under heavy traffic conditions.

When planning for bandwidth consumption using a 28.8K modem, you should actually plan for a maximum rate of 24.6K. The G.723 audio codec will use 6.3Kbits/sec in one direction not including overhead. In general with overhead, it is closer to 10Kbits/sec, but could be higher depending upon other factors. The available bandwidth is adequate for audio and data assuming that audio is only flowing in one direction. When data throughput is high, audio quality may be temporarily degraded. It will become worse when two people are speaking at the same time. H.263 video consumes 16Kbits/sec not accounting for overhead. Therefore, video is not recommended for use on a 28.8K modem.

Codec. A codec compresses streaming data, such as audio or video, on the transmit side and decompresses it for playback on the receive side. The reason for using a codec is to reduce the bandwidth required to send streaming data. However, in order to achieve higher compression, quality is sacrificed as well as using more CPU cycles.

In the case of audio, compression essentially removes unnecessary sound components. For instance, voice resides in a specific frequency spectrum. Music resides in a much broader frequency spectrum. An audio codec, such as G.723, works well for transmitting voice, and is able to compress the data by removing all frequencies except for voice. This is why music sounds poorly when played through the G.723 codec, but voice is acceptable.

Duplex. Duplex describes the mode in which audio is transmitted and received during a meeting. When you have a telephone conversation using a standard telephone handset, you are conversing in full-duplex mode. This means that while you are speaking, you can also hear the other person talking. This is a more natural conversation form in that you can begin speaking at the end of the other person's remark, or even interrupt that person. Full-duplex mode is unlike a speakerphone, which operates, in half-duplex mode. When using a speakerphone, you cannot hear anything that the other person is saying while you are talking.

The reason for this is feedback. If a speakerphone's microphone and speaker were active at the same time, the output from the speaker would feed back into the microphone. If the other person were using a handset, everything that person said would be echoed back. This would be annoying. In addition, if the other person were using a speakerphone, a feedback loop would be created causing a high-pitched squeal. To prevent this, the speakerphone only activates the microphone when it detects that you are speaking, and then it disables the speaker.

Echo Cancellation. Echo cancellation is a technology used in full-duplex mode to prevent feedback. When two audio signals are 180 degrees out of phase, they cancel each other when combined. Echo cancellation software or hardware takes the audio output, inverts the phase by 180 degrees, and combines it with the input audio signal. This signal is then sent on to the remote party. This technology allows a device like a speakerphone to operate in full-duplex mode without feedback.

Jitter Buffer Management. Jitter buffer management occurs on the receiver side. Audio packets are streamed at a consistent rate. However, due to varying network conditions, the arrival time of packets is unpredictable. Some packets may arrive exactly as expected, while other packets may take longer to arrive. In addition, some packets

may arrive out of order or even be lost. The jitter buffer will save several packets before playback occurs.

For example, if each packet contains 30ms of audio, the jitter buffer may save up to 10 packets. This adds 300ms of latency to playback, but this significantly increases the overall quality. In a broadcast situation where no interaction is required, a jitter buffer size of 10 to 30 seconds may be more common.

Latency. When two audio clients are connected in a point-to-point meeting, and one person speaks, a substantial amount of time can pass before the other person hears what was said. This delay is known as latency. Latency is caused by several factors: endpoint processor performance, network bandwidth utilization, and jitter buffer management.

Before data received from the microphone can be transmitted onto the network, the audio must encode this data into the proper negotiated codec form. Encoding audio data can consume 15 to 30 times more CPU cycles than decoding the same audio. This is why a fast processor with MMX is recommended (see MMX below). A processor with MMX can help reduce the amount of time between speaking and actually putting the data on to the wire. When data is received at the other endpoint, the data must be decoded before it can be played out the speakers. This accounts for some delay, but not nearly as much as the sender. DirectX can greatly reduce the amount of processing time required between the network and the microphone or speakers.

Available network bandwidth can be the largest contributing factor to latency. It takes time for a packet to travel between two endpoints. If that pipe is full, the packet can either be delayed or discarded (see Jitter Buffer Management). Either situation causes increased latency and even garbled speech. If an endpoint is using a 28.8K modem connection, video should not be used. The bandwidth requirements are too high, resulting in poor audio quality. Data sharing can also affect audio over a slow connection.

MMX. MMX is a set of multimedia extensions, which Intel first added to their Pentium processors. In much the same way that a numeric co-processor provides software with better number-crunching performance, MMX provides a codec with a significant performance improvement in encoding and decoding streaming data.

RTP. Real-time protocol for sending media packets such as audio or video. Each packet contains among other things a payload type, sequence number, and timestamp. The packet also contains the codec data as the payload.

RTCP. Real-time control protocol for sending reports containing statistics about number of packets received, packets lost, and timestamps for synchronization purposes. It can also contain transmission statistics from the sender, as well as canonical names associated with media sources.

SIP. Session Initiation Protocol, an IETF sponsored standard, is gaining acceptance as an alternative to H.323 in IP telephony for call establishment and call signaling. However, for the purposes of "initiating" a session (or meeting), the protocol is significantly simpler than corresponding H.323 signaling.

Unlike H.323, which requires the use of ASN.1 PER (a relatively complex binary encoding), SIP uses simple text-based messages. Encoding and decoding SIP messages is trivial. What this means is that a client will require less code to process the basic call signaling messages.

UDP. Audio and video streams travel between endpoints using an unreliable connectionless form of IP communications known as UDP (user datagram protocol). Unreliable means that the packets are not guaranteed to arrive in order and possibly may not even arrive at all. Connectionless means that an active connection between the two endpoints is not maintained. If one endpoint goes away, the other endpoint will not be notified of this event.

3 MMCU

3.1 Overview

The MMCU is primarily responsible for managing the control channels between the client and server. When a client establishes a connection to a Watson server, the MMCU will inform the client about meeting attributes such as whether or not video is present for this meeting and which codecs are to be used.

The MMCU registers with the Watson event system in order to monitor meeting events. In addition, the MMCU attaches itself to MCS32 interface of the T.120 stack. This T.120 interface allows the MMCU to communicate with the clients via SIP as well as with the roster manager for conductorship and attributes such as "who has the microphone".

As we move to the places architecture, the roster manage will no longer be used. We still need to determine the new interface to the MMCU to replace the roster manager functionality. As an aside, we are also investigating having the MMCU use Thin MCS rather than MCS32. The MMCU is one of the last components in Watson utilizing MCS32. Removing this interface would benefit any future porting efforts.

The MMCU is interested in knowing when a meeting goes active and when it finishes. The MMCU also watches for the Watson server shutdown event. When a meeting goes active, the MMCU reads the associated meeting document within the Watson Meeting Center. From this document, the MMCU determines whether or not a meeting contains audio. Any meeting not containing audio is ignored by the MMCU.

Video only meetings are not supported by the MMCU. All meetings serviced by the MMCU must at least contain audio.

Once the MMCU determines that an active meeting contains audio, the MMCU will create an associated meeting object for that meeting. This meeting object will contain attributes as defined in the meeting document obtained from Meeting Center database as well as global server settings as defined by the system administrator. The following list outlines some of the more significant attributes associated with a meeting:

- Audio Codec (G.711 or G.723)
- Video Codec (H.263)
- Video Frame Rate

- Video Bit Rate (16 Kbps to 128 Kbps)
- Permit / Deny H.323 endpoints (H.323 endpoints cannot be authenticated)
- H.323 Meeting Identifier (used by H.323 endpoints to join meeting)
- Encryption Enabled

Once the meeting object is created, the MMCU will then create an instance of a media stream object within the MMP – one for each media stream type. For instance, if a meeting has both audio and video, two media stream objects will be created – one to handle audio and the other to handle video. When a meeting ends, the MMCU will shutdown all active client connections.

3.2 Client Connections

The MMCU currently supports two types of client connections: SIP and H.323. The Watson Meeting Room Client uses the SIP connection method. Clients such as Microsoft NetMeeting use the H.323 connection method. The SIP connection method does not currently accept connections for SIP clients other than the Watson Meeting Room Client.

Even though SIP is in the process of becoming a standard, we use it in a proprietary fashion. The Watson Meeting Room Client uses SIP over the Thin MCS connection rather than establishing a direct connection the MMCU as a 3rd party SIP client would expect. When SIP clients become available, the MMCU will support these clients in a fashion similar to H.323 clients. This is not a planned feature for Watson.

3.2.1 SIP

The Watson Meeting Room Client communicates with the MMCU through the Thin MCS connection. In other words, for data, audio, and video, the client has a single connection to the Watson Server.

During the initial connection, the MMCU informs the client of the various attributes associated with this meeting. Most importantly, the MMCU will inform the client which codecs to use for the meeting as well as any parameters necessary to control the codecs. One example of control parameters would be the associated frame and bit rate for video.

Another important exchange of information between the client and server is the ports that will be used for media. When a client establishes a connection, the MMCU will create a client object within the MMP stream object. The MMP stream object will return a pair of port numbers. One is for RTP and the other is for RTCP. These port numbers are passed back to the client via SIP. When the client outputs audio or video, it will use these port numbers to send the media data to the server. Keep in mind that there is a pair of port numbers for each media stream type. The client will also provide to the server a pair of port numbers for each media stream type on which it expects to receive media data. The MMCU passes these port number pairs to the MMP for the associated media stream object.

3.2.2 H.323

The MMCU supports connections with H.323 endpoints. The MMCU listens for H.323 clients on port 1720. Once an H.323 client connects using the Q.931 protocol standard, the E.164 field is examined to determine which meeting the H.323 client wishes to join. This meeting identifier is like a telephone number for the meeting. During the creation of the meeting, if H.323 clients are permitted to join the meeting, a unique H.323 meeting identifier must be assigned to the meeting. If the meeting does not allow H.323 clients to be joined, the MMCU will terminate the client connection.

Once it has been determined the meeting to be joined, the MMCU will transmit a dynamic TCP port according to Q.931 in order for the client and server to continue the call setup using the H.245 protocol standard. It is during the H.245 setup phase that capabilities are exchanged and agreed upon between the client and server. The server does not allow the connection to continue unless the client abides by the codec and attribute settings for the meeting. The final step in the H.245 setup is the exchange of media RTP and RTCP ports. This is the same procedure as the Watson Meeting Room Client and SIP.

Not only are H.323 clients supported in this fashion, but also PSTN users can call into a Watson meeting through an H.323 gateway.

3.3 MMP Events

The MMCU receives certain events from the MMP. For instance, when the audio switches from one client to another client, the MMCU receives an event as to which client is the active presenter. The MMCU will then set the speaking state to TRUE for that participant in the T.120 roster manager and sets the speaking state to FALSE for the previous active presenter. This interface will change somewhat when the places architecture is introduced and the T.120 roster manager will no longer used.

When the active presenter changes, the MMCU will make a call into the MMP to set the video source to the active presenter provided that video is available for the meeting. If the active presenter does not have video or the video has been paused, the MMCU will send an event to all clients that the received video window should be paused. This is either done through SIP for the Watson Meeting Room Client or through H.245 for H.323 clients.

In order to support conductorship, the MMCU will notify the MMP as to which clients have permission to source audio. For those client connections that do not have permission, the MMP will ignore the input audio stream.

4 MMP

4.1 Overview

The MMP is responsible for managing the media streams. All media streams flow from the client to the MMP and then back to the clients. All of the switching logic is handled by the MMP.

The MMP interfaces with the MMCU via DCOM. The purpose of this is so that the MMP can be distributed on to different servers separate from the Watson server. It is possible for an MMP to be running on many different machines all servicing one Watson server.

For each meeting, the MMCU will instantiate one MMP object for each media stream. For a meeting that has both audio and video, two MMP objects will be instantiated.

4.2 Sequence Numbers

Each RTP packet has a sequence number as part of the RTP packet header. The purpose of the sequence number is so that the client receiving the packets knows the proper order in which to play back the packets. When packets are sent using UDP, it is possible for packets to be received out of order. It is also possible that packets may be lost before being received. This is the nature of UDP.

Because the MMP will switch between many different clients during the course of a meeting, the MMP needs to make sure that sequence numbers remain consistent for the receiving client. In other words, the MMP cannot merely send the sequence number from the sending client to the receiving client.

The MMP maintains the last sequence number sent to each client. The MMP also maintains the last received packet number from each client. When a client connects to the server, these values are initialized to zero. As packets are received and sent, the sequence numbers are updated for each client.

For example, three clients, A, B, and C, are currently connected to a meeting. Clients B and C have been in the meeting for a while prior to Client A. The outbound sequence number for Client B is 100 and Client C is 125. Client A has become the active presenter. The following table shows the outbound sequence numbers for Clients B and C that are dependent upon the inbound sequence number received for Client A.

Client A (Inbound)	Client B (Outbound)	Client C (Outbound)
4	104	129
1	101	126
2	102	127
3	103	128
5	105	130
6	106	131
8	108	133
7	107	132
9	109	134

In the above example, sequence number 4 was the first packet received from Client A. Because the value of the last packet received from Client A was initialized to zero, the difference is four. The value of four is then added to the outbound sequence number for Clients B and C. For the next packet, sequence number 1 is received from Client A. The difference between this packet and the last packet received is –3 (1 – 4). Therefore, the next sequence number of Clients C and B is the last sequence number sent minus three.

4.3 Timestamps

Just like sequence numbers, each RTP packet header has a timestamp. The client places this timestamp in the header. The receiving client uses this timestamp to determine the proper time to play back the packet. As with sequence numbers, the server

must maintain the outbound timestamp. The server cannot simply pass the inbound timestamp to each outbound stream because as the source switches between inbound streams, that timestamp value would be meaningless to the outbound client.

The actual value of a timestamp is dependent upon the type of data in the payload. For instance, in the case of the G.723 audio codec, the value of the timestamp is based upon 240 samples per frame. Each frame of data contains 24 or 20 bytes of data depending upon the encoding method and equates to 30ms of time. If audio is being continuously streamed and there is one frame per packet, the value of the timestamp will increase by 240 for each packet.

When the MMP switches to another inbound stream, the MMP will store the value of the current server. Then, as each new inbound packet is sent to the outbound clients, the value of the stored server time will be incremented by the difference (either positive or negative depending if packets are received out of order) of the current and last received packet of the inbound client. This server time value will be used for all outbound clients. In other words, unlike the sequence numbers, all outbound packets contain the same timestamp value and are based upon the server time. When the first packet is sent after a switch occurs, the marker bit is set. This is to let the clients know that the timestamp value has been adjusted so the clients can handle this correctly in regards to their jitter buffer management.

Instead of resetting the server time each time a switch occurs, the timestamp value could have remained relative from the start of the meeting based upon the timestamps received from the inbound clients. However, because clocks can drift, especially across many different clients, it was decided that it is best to readjust the time between each switch. Depending upon how a receiving client handles this, audio quality could degrade for a short period of time for that client. However, the client should adjust correctly if it is watching for the marker bit being set. This would generally only be a problem if the client did not receive the first packet after the switch with the marker bit set because the network dropped the packet.

4.4 Full Duplex Mode

The MMP supports the concept of full-duplex mode. This occurs when there are only two clients in a meeting. In this case, the audio and video stream from one client is sent to the other client and vice-versa. The MMP will still handle the sequence numbers and timestamps in the same way that it does for non-full duplex mode.

4.5 Two-Way Mixing and Push-To-Talk Mode

The MMP does not understand the difference between two-way mixing and push-to-talk mode. This is strictly a client convention. In order for push-to-talk mode to work, it is up to the clients to manage themselves so that no more than one client is actively sourcing audio at a time. This also means that H.323 clients cannot be supported in push-to-talk mode.

4.6 Multiple Audio Inbound Streams

The MMP will lock on up to two inbound audio streams. One stream will be designated as primary and the other as secondary. The primary stream is the stream that has been sourcing the longest. This does not mean the stream that has been connected the

longest. As the primary stream stops sourcing and the secondary stream continues to source, the secondary stream will become the primary stream.

The MMP will send both the primary and secondary audio streams to all clients capable of receiving two streams. In the case of H.323 clients, only the primary stream will be sent. The only exception to this is that if an H.323 client is the primary or secondary source, that H.323 client will receive the other stream if available.

5 H.323 Gatekeeper Support

The H.323 gatekeeper strategy for Watson still needs to be determined. Because Watson will interact will H.323 clients and gateways, Watson should provide some minimal support of H.323 gatekeepers. At a minimum, Watson will probably need to register with a gatekeeper as an MCU and gateway along with an H.323 prefix. The H.323 prefix is much like defining an area code. For instance, if Watson has registered with a gatekeeper using a prefix of 8 and the H.323 meeting identifier is 1234, an H.323 endpoint can connect to the correct meeting through a gatekeeper by calling "81234" which is a combination of the prefix and meeting identifier.

6 Bandwidth Management

In an H.323 environment, a gatekeeper controls the bandwidth that is used by H.323 entities on the network. It does this be accepting or rejecting calls based upon the amount of bandwidth requested for the call and what is permitted by the network administrator. Watson will not support the use of a gatekeeper for bandwidth management of the Watson Meeting Room Client. However, there is no reason why a network administrator should not be able to use a gatekeeper to perform bandwidth management for H.323 endpoints used in conjunction with Watson.

Watson needs a more robust control method for bandwidth management. The facility offered by the H.323 gatekeeper works well in a dynamic environment where one person calls one or more others similar to a phone call. However, in Watson, many meetings are scheduled ahead of time so these resources need to be reserved in advance. Even though Watson does not actually perform port reservation today, it will in future versions.

In order to perform bandwidth management for Watson when used with the Meeting Room Client, a Watson server limits the number of active connections. If the administrator wishes to allow no more than 20 active clients at any given time for a particular server, the administrator would use the administration facility to specify the maximum number of active audio and video ports. In order for a network administrator to control bandwidth usage by a Watson server, the network administrator needs to understand the bandwidth used by specific codecs.

6.1 G.723 Audio Codec Bandwidth Usage

Packetization, in accordance with RFC 1890, and overhead introduced by the RTP protocol, described in RFC 1889, the Internet Protocol (IP), described in RFC 791, and the User Datagram Protocol (UDP), as described in RFC 768, necessarily increases the bandwidth utilization for a single port significantly. A single media flow (i.e., an audio stream transmitted from one endpoint to another endpoint) requires approximately 17.1

Kbps of bandwidth when one audio frame is transmitted per RTP packet, which is how Watson currently operates. By comparison, if three frames were encoded per packet, the bandwidth utilization would be approximately 9.95 Kbps. These values were calculated as follows:

G.723.1 frame size (s):	24 bytes
Frames per packet (f):	1
RTP header (r):	12 bytes
UDP header (u):	8 bytes
IP header (i):	20 bytes
G.723.1 frame rate:	1 frame / 30ms

The general form of the equation to calculate bandwidth usage, also referred to as "session bandwidth" in RFC 1889, is:

$$Kbps = \frac{(s*f+r+u+i)\,bytes}{1\ packet} * \frac{8\ Kbits}{1000\ bytes} * \frac{1\ frame}{30\ ms} * \frac{1\ packet}{f\ frames} * \frac{1000\ ms}{1\ s}$$

To understand the bandwidth utilization for the entire server, it is important to understand how media is transmitted to participants in a meeting. Refer to the MMP for more information about how media actually flows to and from the server.

In a two-person Watson meeting, media is transmitted from the two endpoints to the server. The server then transmits the two streams to the opposite party -- making a total of four streams on the network. The maximum bandwidth requirements for a two-person meeting is then:

$$(2\ inbound\ streams + 2\ outbound\ streams) * 17.1 Kbps = 68.4 Kbps$$

In a meeting with three or more people, each person may transmit a stream and the server will transmit one stream for the primary speaker and possibly a second stream of a secondary speaker to all other endpoints. In that scenario, there are n inbound streams to the server and up to (n-2)*2+2 outbound streams from the server. In that scenario, the total bandwidth that may be used for a five user meeting, for example, is (5 + (5-2)*2+2)*17.1 = 222.3Kbps. The general form of the equation for calculating the maximum bandwidth utilization for an n-person meeting is:

$$(n\ inbound\ streams + (n-2)*2 + 2\ outbound\ streams) * 17.1 Kbps$$

It is important to understand that these bandwidth statistics represent a maximum. The Watson client and server will not transmit audio when silence is detected. During a typical phone conversation, there is only one audio stream flowing to the server and one stream flowing out of the server to each of the other endpoints. So in reality, the typical bandwidth usage for an n user meeting is n*17.1. Given that, the bandwidth utilization for the five person meeting mentioned above would typically be 5 * 17.1 = 85.5Kbps. When determining bandwidth requirements, it should be understood that 85.5 would be the typical and that 222.3Kbps would be the maximum usage for a five-person meeting. The

general form of the equation for calculating the typical bandwidth usage for an n person meeting is given by this equation:

$$(1\ inbound\ stream + n-1\ outbound\ streams) * 17.1 Kbps$$

Suppose that an administrator chooses to specify the number of audio ports to be 20 ports. This will instruct the server to allow a meeting with up to 20 participants or a number of smaller meetings whose total participant count in all meetings is 20 or fewer. The maximum bandwidth utilization for a 20-person meeting would be (20 + (20-2)*2 + 2)*17.1 = 991.8Kbps, while the typical 20-person meeting will actually consume 20 * 17.1 = 342Kbps. In addition, ten two-person meetings would require (2 inbound streams + 2 outbound streams) * 17.1 * 10 meetings = 684Kbps at a maximum and (1 inbound stream + 1 outbound stream) * 17.1 * 10 meetings = 342Kbps typically.

During periods where nobody is talking at all, which may actually be more frequent than one might think, the bandwidth utilization will be little or nothing at all. So even the numbers given above for a "typical" meeting are probably higher than "typical". There may be periods of silence, which consumes nearly 0Kbps of bandwidth.

In addition to the bandwidth requirements stated in the above paragraphs, each RTP stream has an associated RTCP stream. The total RTCP traffic for any given RTCP stream may be as much as 5% of the bandwidth utilization for the corresponding RTP session. In other words, if there are 10 participants in a meeting consuming 348 Kbps, it is possible that an additional 17.4 Kbps may be utilized by RTCP. With this in mind, the maximum bandwidth consumption for "n ports" is roughly:

$$((n\ inbound\ streams + (n-2) * 2 + 2\ outbound\ streams) * 17.1 Kbps) * 1.05$$

However, given that the maximum is not the typical meeting scenario, bandwidth for "n ports" should probably be estimates as follows:

$$((1\ inbound\ stream + n-1\ outbound\ streams) * 17.1 Kbps) * 1.05$$

The above examples should probably be reworked using 3 frames per packet instead of one frame per packet, which will significantly reduce the amount of required bandwidth. It is also worth noting that the above examples do not take into account the fact that there may be two active presenters. In this case, two streams are sent to each client except for the two presenters whom will receive each other's stream. For H.323 clients, they will only receive one stream. However, it is a rare occurrence that two presenters will be active for long periods of time. This usually only occurs as an interruption and transition from one presenter to another. Bandwidth usage will increase for a short period of time during this condition.

6.2 G.711 Audio Codec Bandwidth Usage

TBD…

6.3 H.263 Video Codec Bandwidth Usage

TBD…

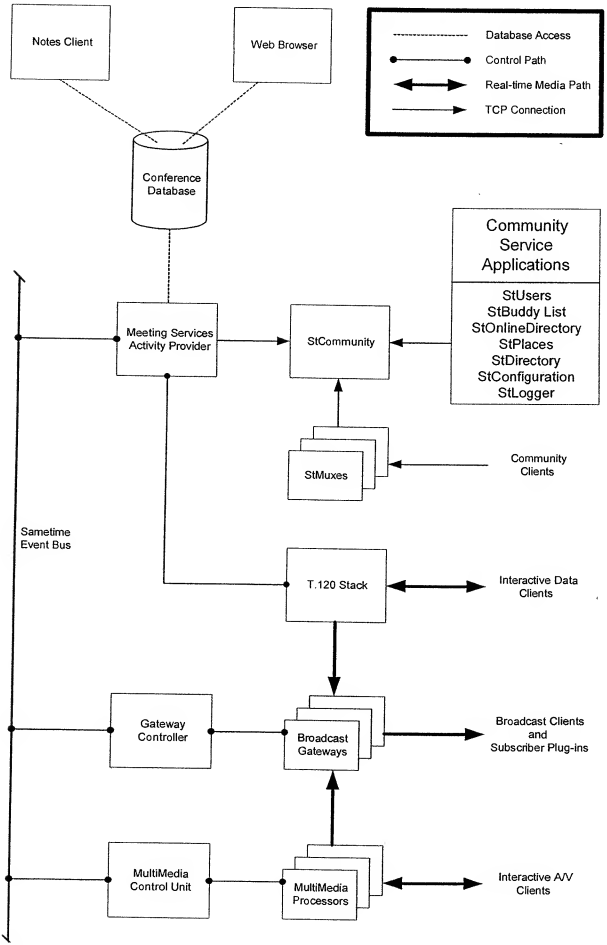
7 Invited Server

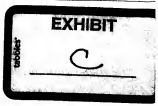
For Watson, audio and video will not behave like data for invited servers. For data, clients connect to their local servers. For audio and video, clients will need to connect with the top provider. The local server provides data, but the client will be redirected to the top provider for the meeting for audio and video.

The overall design of invited servers for audio and video is still TBD…

8 A/V Tuning Wizard

The A/V Tuning Wizard utilizes the MMCU and MMP to test both audio and video connectivity and the client mixer configuration by creating a special type of meeting that reflects both audio and video received by the MMP back to the client. This special meeting type has at most a single participant. In this scenario, the MMP simply reflects the A/V stream received from the endpoint back to the sending client. The client then plays back its own audio and video allowing the endpoint to completely test the entire audio and video chain including encoders, decoders and network connectivity.


Notes Client
Web Browser
Database Access
Control Path
Real-time Media Path
TCP Connection
Conference Database
Community Service Applications
StUsers
StBuddy List
StOnlineDirectory
StPlaces
StDirectory
StConfiguration
StLogger
Meeting Services Activity Provider
StCommunity
StMuxes
Community Clients
Sametime Event Bus
T.120 Stack
Interactive Data Clients
Gateway Controller
Broadcast Gateways
Broadcast Clients and Subscriber Plug-ins
MultiMedia Control Unit
MultiMedia Processors
Interactive A/V Clients



From: <Brian_Pulito@DataBeam.com>
To: BK.BK_PO(bjobse)
Date: [redacted]
Subject: Design Documents related to 2-way mixing patent

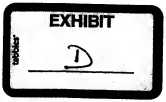
Bruce,

Attached are the design documents you requested. Note that there is probably more info here then you actually need but these documents describe how the server side media components work. Let me know if there is anything else I can do to help.

Thanks,
-brian

(See attached file: Watson MMCU MMP Design Specification.doc)(See attached file: Sametime Server Overview Diagram.doc)

CC: BK.gwia("Steve_Keohane@DataBeam.com")



From: <Brian_Pulito@DataBeam.com>
To: BK.BK_PO(bjobse)
Date: Mon, [redacted] 4:04 PM
Subject: Other Contact Info for MCU Patent

Mark Johnson: 513-772-4566 x12
Brian Cline: 513-772-4566 x17
Mark Kressin 512-261-0165